

*MBL317*



# *mobile PC & devices*

Under the Covers: Windows  
Mobile Device Platform  
Futures

Ori Amiga  
Lead Program Manager  
Mobile Devices Product Group  
Microsoft Corporation

**PDC**<sup>03</sup>

Make the connection

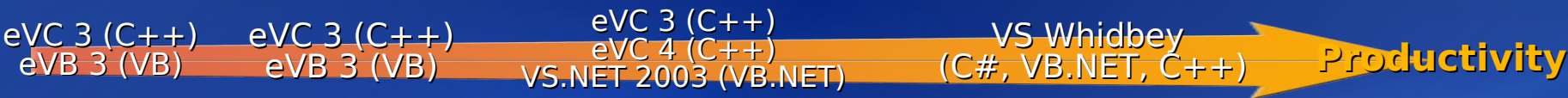
**Microsoft**<sup>®</sup>

# Agenda

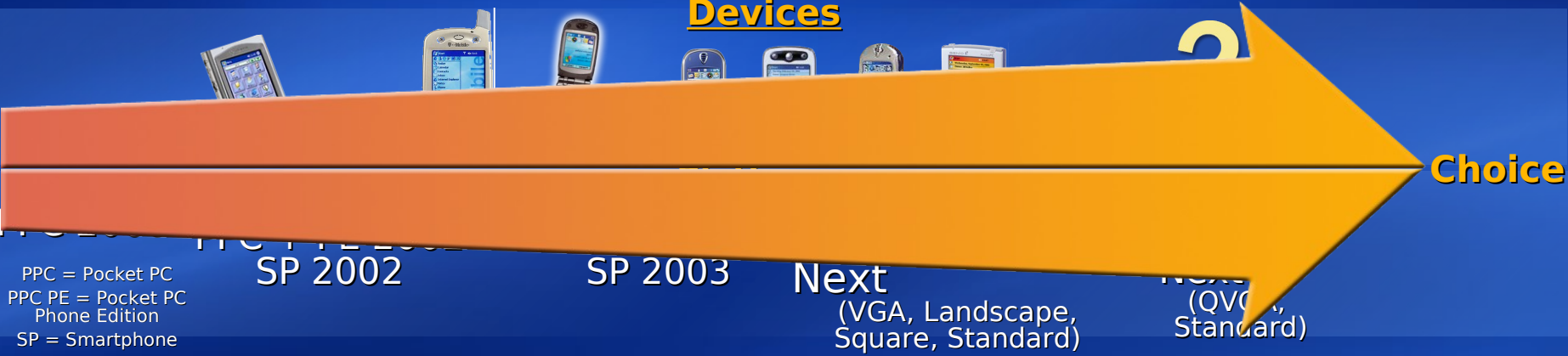
- Platform Evolution
- Demo: Flight Service (C#)
  - Notifications Broker
  - Managed access to Calendar
- Demo: Contacts and SMS (C#)
  - Managed access to Contacts database and data-binding
  - Sending an SMS message
- Demo: Making a Phone Call (C#)
  - Making a call using new Telephony namespace
- Demo: Email and Camera (C#)
  - Creating an Email form
  - Using the Camera API to attach a photo
- Demo: Configuration System (C#)
  - Access the Windows Mobile XML configuration system
- Demo: Location-Based Services (C#)
  - Using the Location platform APIs to retrieve current device location
  - Leveraging live Web Services in your application
- Demo: DirectX Mobile
  - Quick n' dirty application using Direct3D Mobile

# Windows Mobile Roadmap

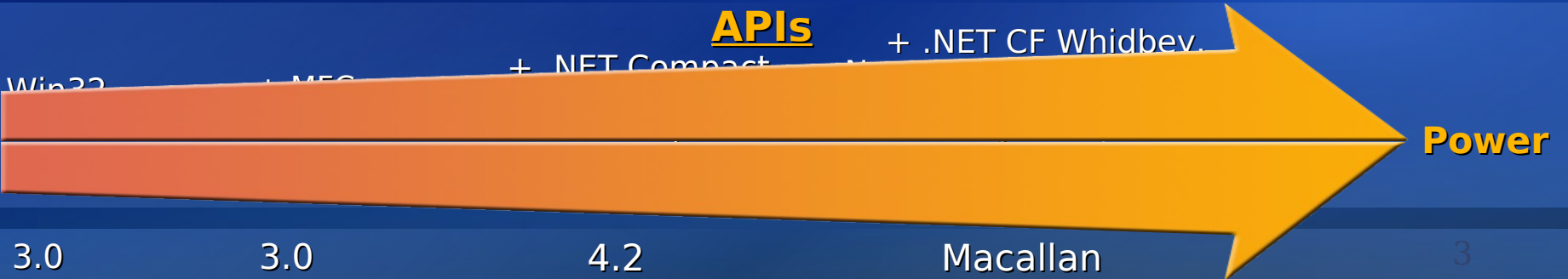
## Tools



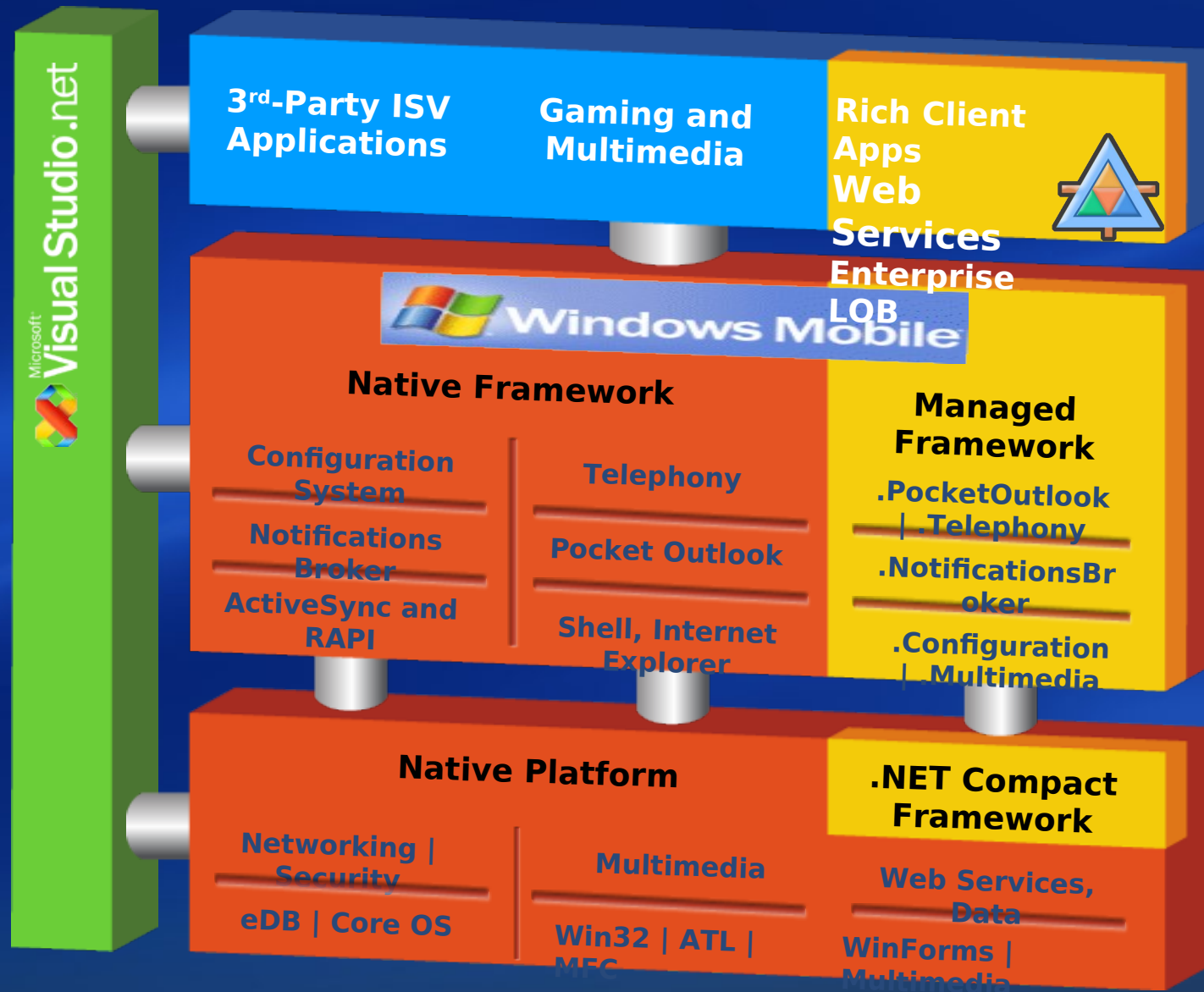
## Devices



## APIs



# Future Windows Mobile Platform



# Notifications Broker

“Notifications-driven” code is a key programming paradigm of Pocket PC and Smartphone applications.

- Key benefits:
  - Improved context-sensitive functionality
  - Tighter integration with system services
  - Efficient use of CPU, memory, power, bandwidth
  - Extensible by OEMs and ISVs
- Today:
  - Limited notifications support
  - Multiple different models for different components
  - Limited/no support for .NET applications notifications and activation.
- Tomorrow: notify via:
  - Window Messages
  - CE Message Queues
  - Application Activation
  - .NET Event Delegates
- Unified notifications on:
  - Networking state
  - SMS / Email
  - Synchronization state
  - Telephony events
  - Device state changes
  - 3rd-Party-defined events
  - And more.



# Notifications Broker

# demo

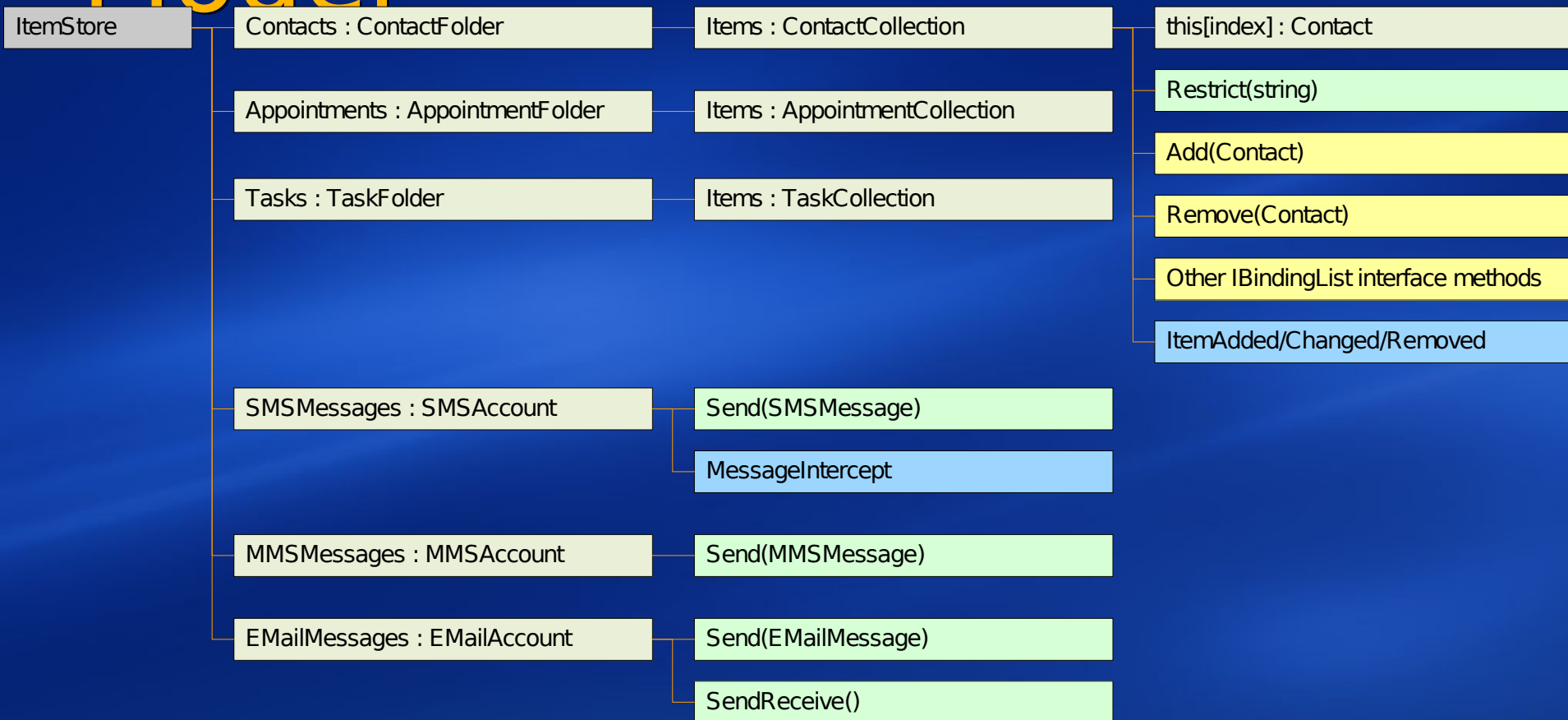
**PDC**<sup>03</sup>

Make the connection

# Pocket Outlook Object Model

- Today: separate unique native APIs for each core area
  - ▣ Native POOM
  - ▣ Native MAPI
  - ▣ Native SMS API
  - ▣ Native SMS intercept
- Tomorrow: unified .NET programming model
  - ▣ Mail
  - ▣ SMS
  - ▣ Contacts
  - ▣ Calendar
  - ▣ Tasks
- Supporting (common):
  - ▣ Creation/Deletion
  - ▣ Enumeration
  - ▣ Sort/Search/Restrict
  - ▣ Copy/Compare
  - ▣ Data binding

# Pocket Outlook Object Model



Properties

Methods

Interface implementations

Events



# Contacts and SMS

# demo

**PDC**<sup>03</sup>

Make the connection

# Telephony Object Model

- Today: separate unique native APIs for
  - TAPI
  - ExTAPI
  - Assisted TAPI
  - Phone API
  - Call Log
- Tomorrow: unified .NET programming model
  - Phone control
  - Call log access
- Supporting:
  - Access telephony UI
  - Make / end calls
  - Determine phone state
  - Access call log
  - Notifications

# Making a Phone Call

# demo

**PDC**<sup>03</sup>

Make the connection

# eDB

## Future Generation eMbedded Database

- eDB is the replacement data store for CEDB
- New features:
  - ACID transactions
  - Significantly better performance with large data sets
  - Larger databases
  - Support for more indexes
- Replacing the PIM data store
- CEDB-like API, not 100% compatible
- Supports a secure file system  
(i.e. encryption at the file level with password)

# Mail and Picture Control demo

**PDC**<sup>03</sup>

Make the connection

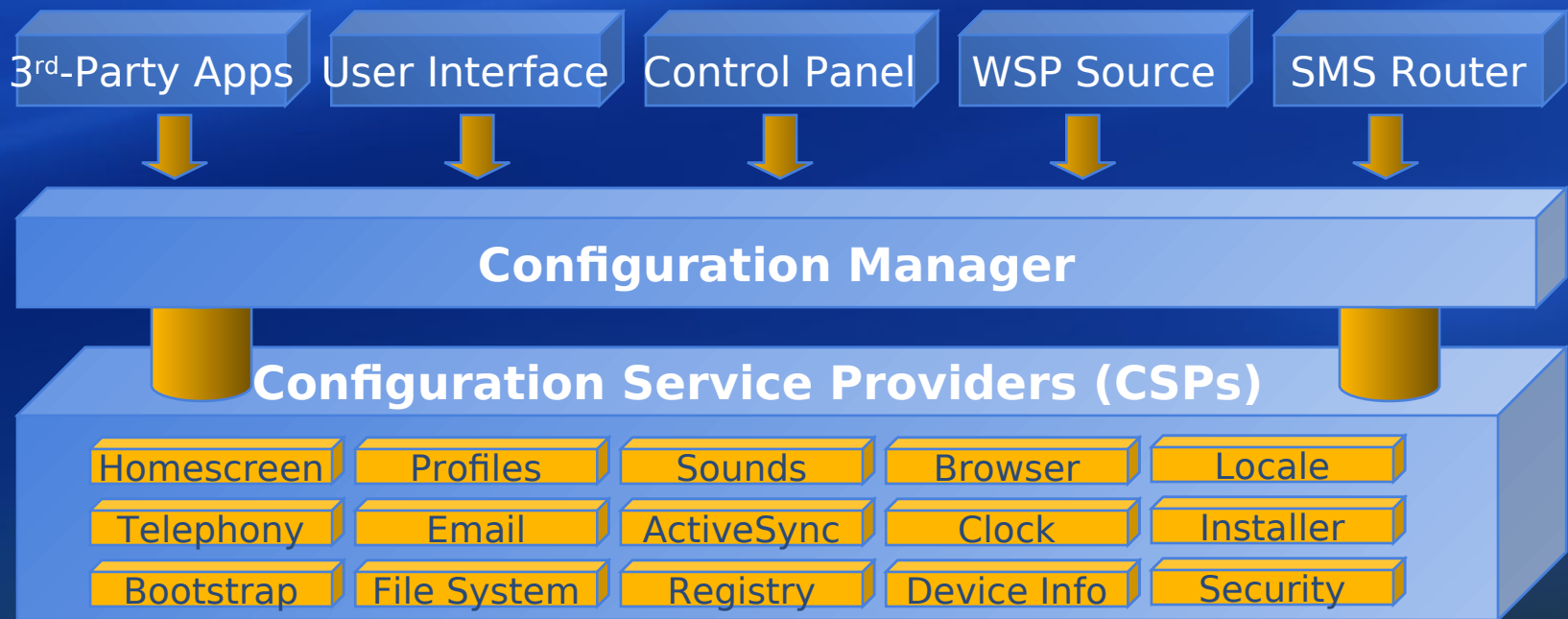
# Configuration System

## ● Config Manager

- Single point of configuration
- Brokers requests to CSPs
- All operations transacted
- Consistent security role across transactions

## ● Provisioning Supported Via:

- WAP Push (OTA)
- CAB-Based Provisioning (.CPF)
- DMProcessConfigXML API
- .MobileDevices.Configuration
- RapiConfig.exe from Dekstop





# Configuration System Access

# demo

**PDC**<sup>03</sup>

Make the connection

# Platform Evolution:

## Hardware

- Displays

- Pocket PC Portrait/Landscape support
- Pocket PC Square Screen
- Pocket PC VGA
- Smartphone QVGA

- Form Factors

- Hybrid devices
- Pocket PC softkey support

- Multiple-Radios

- GSM/GPRS CDMA/1xRTT
- WiFi (802.1x)
- Bluetooth
- Location Services

# Location Service Architecture

- **Location Data API**

- Applications register for location notifications
- Applications retrieve location reports

- **Management API**

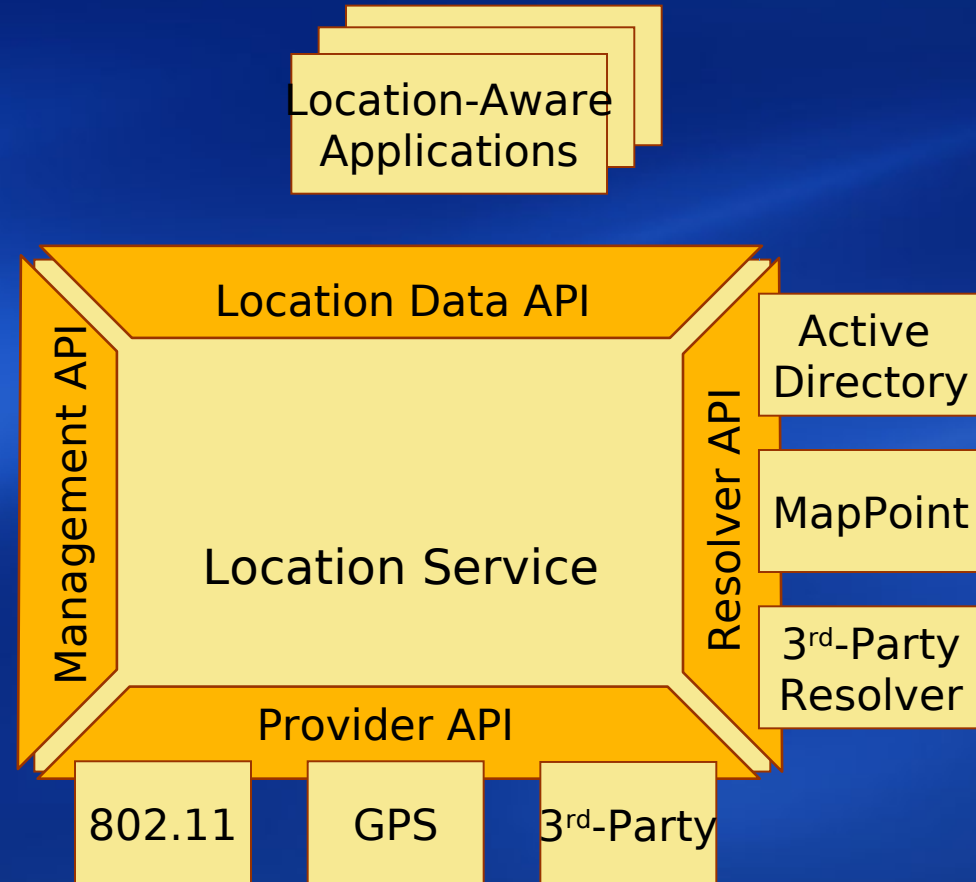
- Applications manage subsystems of service

- **Provider API**

- Feed location information into the service

- **Resolver API**

- Receive location reports from the



# Location and Web Services

# demo

**PDC**<sup>03</sup>

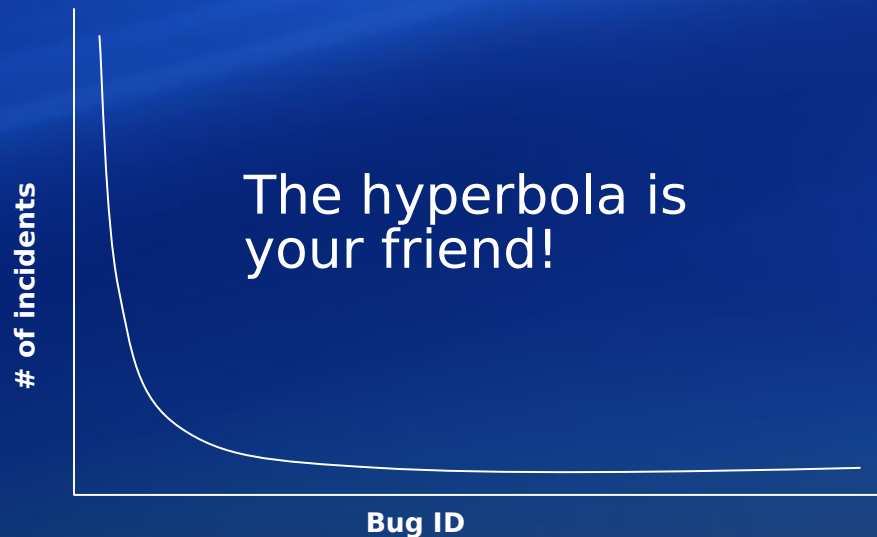
Make the connection

# Platform Components

- Updated Web Server (HTTPD) w/ ASP support
- Native MSMQ libraries, as well as managed support for System.Messaging
- SOAP Toolkit to consumption of web-services in native code
- UPnP
- P2P
- Updated version of MSXML

# Pocket Watson

- Testing catches a large number of bugs but conditions in the field are difficult to emulate
- Top ~1% of bugs cause ~50% of the user-encountered crashes



- Pinpoint problem areas that users are actually experiencing “in the field”
- Report crash data with the option of including other logs
- Leverages existing Desktop Watson reporting infrastructure
- Complies with US and EU principals on collection and use of personal data
- Reported directly from device
- Report primarily via non-cellular data connections (avoid charges)
- ISV's who participate in Mobile2Market can access their crash data via the Windows Error Reporting Program (WERPA)



# Multimedia

- Direct3D Mobile
  - Proper subset of desktop Direct3D 8 API (COM based)
  - Designed from ground up targeting devices (small code-size and footprint)
  - Flexible Architecture (SW/HW/Hybrid)
  - Support for both native (Win32) and managed (.NETCF) on Pocket PC and Smartphone
- DirectDraw
- DirectShow (camera and video image capture)
- Windows Media Player
  - Pluggable Codecs
  - OGCY control

Quick n' Dirty: D3DM

demo

**PDC**<sup>03</sup>

Make the connection

# Visual Studio Whidbey

- Managed and Native development in same IDE
- First class development experience
  - WinForms and Data Designers for Managed
  - Resource Editor for Native
  - Accurate Intellisense, integrated Help
  - New device cross-compilers for Native
  - Updated Connectivity technology, native and managed remote debuggers
- Whidbey Native Runtimes
  - ATL, MFC, CSL/STL for native development
  - Bringing device runtimes to desktop 8.0 code base
  - Merging ATL and MFC common functionality
  - Improved security, performance, robustness
  - See MBI 315 on PDC DVD for additional info

# .NET Compact Framework

Whidbey

- Performance
- Extensibility
  - P/invoke
  - COM interop
  - Hosting Interface
- Class Library evolution
  - More GUI controls and drawing
  - Designer support for multiple resolutions
  - More XML (xpath, schemas)
  - MSMQ, Authentication and IPV6 for networking
  - SQL CE result set (fast!)
  - Other: crypto, registry, generics, serial port

# Windows Mobile

## Call to action

- Develop Windows Mobile applications today (Tools available on Longhorn DVD)
- Visit [www.microsoft.com/windowsmobile/developer](http://www.microsoft.com/windowsmobile/developer)
- Join Windows Mobile developer beta programs
  - Mail to : [wmdevbet@microsoft.com](mailto:wmdevbet@microsoft.com)
  - Subject : Windows Mobile Future Gen

# Windows Mobile

## What to do at PDC

- Visit the Product Pavilion
  - See the latest devices and get your questions answered
- Take a Hands on Lab – Pocket PC, Smartphone, MapPoint, .NET Compact Framework
- Visit the Mobile Planet store
  - Buy a Smartphone 2003 Developer Kit
  - Buy a Pocket PC and accessories
- Visit the Mobile PC and devices lounge
  - Comfortable connectivity with experts on hand





# PDC<sup>03</sup>

Make the connection

**Microsoft Professional Developers Conference 2003**

October 26 - 30, 2003, Los Angeles, CA

**Microsoft®**

# Appendi x

# Windows Mobile

## Resources – On the web

- Windows Mobile Developer Portal
  - [www.microsoft.com/windowsmobile/developer](http://www.microsoft.com/windowsmobile/developer)
- Mobility Developer Center
  - [www.msdn.com/mobility](http://www.msdn.com/mobility)
- Mobile2Market
  - [www.microsoft.com/windowsmobile/mobile2market](http://www.microsoft.com/windowsmobile/mobile2market)
- Windows Mobile Community
  - [www.microsoft.com/windowsmobile/resources/communities/developer](http://www.microsoft.com/windowsmobile/resources/communities/developer)
- Smartphone Developer Kit
  - [www.microsoft.com/windowsmobile/information/](http://www.microsoft.com/windowsmobile/information/)

# Windows Mobile Sessions

<b>MBL210</b>	<b>Overview: Developing for Windows Mobile Devices</b>	Mon 1:30-2:45	404AB	James Pratt
<b>MBL314</b>	<b>Developing Data Applications for Devices with SQL Server CE 3.0</b>	Mon 3:00-4:15	404AB	Kevin Collins
MBL315	Developing for Devices Using Native C++ with Visual Studio	Mon 4:45-6:00	402AB	Nishan Jebanasam Rich Hanbidge
<b>MBL319</b>	<b>Building Mobile Location Based Applications with MapPoint Technologies</b>	Tue 2:00-3:15	402AB	Steve Lombardi
<b>MBL317</b>	<b>Under the Covers: Windows Mobile Device Platform Futures (Pocket PC and Smartphone)</b>	<b>Tue 2:00-3:15</b>	<b>511AB C</b>	<b>Ori Amiga</b>
<b>MBL312</b>	<b>Developing Smartphone Applications Using the .NET Compact Framework</b>	Tue 5:15-6:30	402AB	<b>Jamie de Guerre Scott Smith</b>
<b>MBL210R</b>	<b>Overview: Developing for Windows Mobile Devices (Pocket PC and Smartphone)</b>	Wed 3:30-4:45	404AB	James Pratt
MBL311	Exploring New Features in .NET Compact Framework "Whidbey" Release	Wed 3:30-4:45	501AB C	Bruce E. Johnson Seth Demsey